



Windowed total variation denoising and noise variance monitoring

Zhanhao Liu, Marion Perrodin, Thomas Chambrion, Radu S. Stoica

► To cite this version:

Zhanhao Liu, Marion Perrodin, Thomas Chambrion, Radu S. Stoica. Windowed total variation denoising and noise variance monitoring. 2021. hal-03122844

HAL Id: hal-03122844

<https://hal.science/hal-03122844>

Preprint submitted on 27 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Windowed total variation denoising and noise variance monitoring

Zhanhao Liu^{1,3}, Marion Perrodin³, Thomas Chambrion², Radu S.Stoica¹

¹Université de Lorraine, CNRS, Inria, IECL, F-54000 Nancy, France

²Institut de Mathématiques de Bourgogne, UMR 5584, CNRS, UBFC, F-21000 Dijon, France

³Saint-Gobain Research Paris, 39 Quai Lucien Lefranc, F-93300 Aubervilliers, France

Abstract—We proposed a real time Total-Variation denosing method with an automatic choice of hyper-parameter λ , and the good performance of this method provides a large application field. In this article, we adapt the developed method to the non stationary signal in using the sliding window, and propose a noise variance monitoring method. The simulated results show that our proposition follows well the variation of noise variance.

I. INTRODUCTION

The signal $y = (y_1, \dots, y_n) \in \mathbb{R}^n$ collected by the sensor can be modeled as $y = u + \epsilon$: a random noise ϵ is added into the useful physical quantity u with $\mathbb{E}(\epsilon) = 0$ and $\mathbb{V}(\epsilon) = \sigma^2$.

We aim to recover the unknown vector $u = (u_1, \dots, u_n) \in \mathbb{R}^n$ from the noisy sample vector $y = (y_1, \dots, y_n)$ with y_i the sample at time t_i by minimizing the Total Variation (TV) restoration functional:

$$F(u, y, \tau, \lambda) = \sum_{i=1}^n \tau_i (y_i - u_i)^2 + \lambda \sum_{i=1}^{n-1} |u_i - u_{i+1}| \quad (1)$$

with the sampling period vector $\tau = (\tau_1, \dots, \tau_n)$ where $\tau_i = t_i - t_{i-1}$ for $i = 2, \dots, n$ and $\tau_1 = \tau_2$. The restored signal is given by:

$$u^*(\lambda) = (u_1^*(\lambda), \dots, u_n^*(\lambda)) = \arg \min F(u, y, \tau, \lambda) \quad (2)$$

Total Variation based restoration method is first proposed in [1]. The authors in [2] show an one-to-one correspondence between the noise's variance σ^2 and λ under the hypothesis of constant σ^2 .

Motivated by the local influence of new sample to the actual restoration, we propose a new online TV-restoration algorithm with an automatic choice of λ for a stationary 1D signal in [3] following the work of [4] and [5]. The simulations show that our proposition of λ has a similar performance as the existing methods (SURE [6] and cross-validation). Further more, our method is appropriate for real time applications, especially for monitoring a huge amount of sensors in the plants.

In this paper, we adapt our online TV restoration to non-stationary signals and propose a noise variance monitoring method. The key idea is to use our method on sliding windows, assuming a local-stationary hypothesis for the signal. This hypothesis is strong but commonly used already (e.g. Fourier analysis, wavelet transforms). In order to apply our method to non-stationary signals, two problems are to be solved:

- Adaptation of the online method to the sliding window
- Choice of the length of window (noted m)

The first point ensures the efficiency of the real time estimation, while the second point guarantees the good performance of method. Indeed, the choice of m needs to get a compromise between the local stationarity assumption and the performance of the restoration : the local influence of new sample vanishes inside a large window, but the local stationarity may not be respected. In this article, we present how we deal with the first point by supposing the influence of new sample stays inside the actual window. The choice of window size is the immediate perspective of this work.

The increasing of the ground noise is one of the index for the failures of the sensors or the production lines. This motivates us to build a real time application for detecting the variance behaviour by tracking the windowed restoration residuals $y - u^*(\lambda_{ours})$ based on the automatic determination of hyper-parameter λ_{ours} .

Here some approaches for the estimation of noise variance σ^2 are listed:

- Most of the existing methods are based on the separation of signal and noise by digital filters [7] or the restoration methods (i.e. wavelet transform) [8] [9] [10]. See more details in [11].
- Median absolute deviation (*MAD*) proposed by [12] is a heuristic method largely used in many applications, especially for the choice of threshold for wavelet restoration. σ can be estimated by the median absolute deviation of the finest scale of wavelet coefficients By of signal.

With respect to the previous cited method, our approach, based on the separation of signal and noise, proposes a procedure for simultaneously signal denoising and variance tracking, while being implementable as an online application. Thanks to its low time and space complexity, our proposition is well fitted to the cases with limited computation resource.

The paper continues as it follows: in Section II we will at first present our TV-based denosing method with an automatic choice of parameter. Then, in Section III we adapt our online implementation to the sliding window. After that, in Section IV, we will present an application of our algorithms: noise variance monitoring, and compare with an existing method. Finally, conclusions and perspectives are depicted.

II. AUTOMATIC TOTAL VARIATION DENOISING METHOD

In this section, we will present the automatic TV-denoising method proposed in [3]. We aim to recover the unknown vector u from the noisy samples y by minimizing (1). The restored signal is given by (2).

Since we work on a finite sample of signal, the solution $u^*(\lambda)$ can be seen as piece-wise constant. We use the *segment* representation [5] for the constant pieces: a set of index $\{j, j+1, \dots, k\}$ of consecutive points whose restored value $u_j^*(\lambda) = \dots = u_k^*(\lambda)$ is called a *segment* if it can not be enlarged, which means if $u_{j-1}^*(\lambda) \neq u_j^*(\lambda)$ (or $j = 1$) and $u_k^*(\lambda) \neq u_{k+1}^*(\lambda)$ (or $k = n$). The segments number of $u^*(\lambda)$ is noted as $K(\lambda)$. The following notations are introduced for the j^{th} segment:

- Index set $\mathcal{N}_j(\lambda) = \{i_1^j, \dots, i_{n_j}^j\}$ with $n_j(\lambda) = \text{Card}(\mathcal{N}_j(\lambda))$, containing the point inside the j^{th} segment.
- Segment level $v_j^*(\lambda) = u_i^*(\lambda), \forall i \in \mathcal{N}_j(\lambda)$

An equivalent representation of $u^*(\lambda)$ is provided by the couple $\{v^*(\lambda), \mathcal{N}(\lambda)\}$ with the set of *segment levels* $v^*(\lambda) = (v_1^*(\lambda), \dots, v_K^*(\lambda))$ and the *cutting set* $\mathcal{N}(\lambda) = \{\mathcal{N}_1(\lambda), \dots, \mathcal{N}_K(\lambda)\}$. By knowing the cutting set $\mathcal{N}(\lambda)$, let $s(v) = \{s_0(v), s_1(v), \dots, s_K(v)\}$ with $s_i(v) = \text{sign}(v_{i+1} - v_i)$ for $i = 1, \dots, K-1$, $s_i(v) = 0$ for $i \in \{0, K\}$ and $s^* = s(v^*(\lambda))$, the level of j^{th} segment is given by:

$$v_j^*(\lambda) = \bar{y}_j^* + \frac{\lambda}{2\mathcal{T}_j}(s_j^* - s_{j-1}^*) \quad (3)$$

with $j = 1, \dots, K(\lambda)$, the segment length $\mathcal{T}_j = \sum_{i \in \mathcal{N}_j(\lambda)} \tau_i$ and the mean value inside j^{th} segment $\bar{y}_j^* = \frac{\sum_{i \in \mathcal{N}_j(\lambda)} \tau_i y_i}{\mathcal{T}_j}$.

A different value of λ may provide a solution with distinct cutting set: $\lambda = 0$ gives $u^* = y$, while $\lambda = \infty$ implies a parsimonious solution $u^* = \text{mean}(y)$. The authors in [4] and [5] show there exists a sequence $\Lambda = (\lambda_1, \dots)$ such that for every $\lambda \in \Lambda$, two segments are merged together by moving $\lambda - \eta$ to λ with $\eta \rightarrow 0$ and $\eta > 0$. In [3], we propose a rapid algorithm to estimate the dynamic of the restoration in function of λ , and the dynamic is saved in $\Lambda^\circ = (\lambda_1^\circ, \lambda_2^\circ, \dots, \lambda_{n-1}^\circ)$ with λ_i° the value of λ for which the points i and $i+1$ are merged into the same segment. Λ° allows the computation of the cutting set and also the restoration (3) for every λ .

Based on the variation of the extremums number (noted $g(\lambda)$) of the restoration $u^*(\lambda)$ in function of λ , an adaptive choice of hyper-parameter λ is proposed. The simulations show that our estimation λ_{ours} has a similar performance as the state of the art, all near the optimal choice $\lambda_{op} = \arg \min |u^*(\lambda) - u_{net}|^2$ with u_{net} the original signal. The variation $\Delta g(\lambda) = g(\lambda - \eta) - g(\lambda)$ for every $\lambda \in \Lambda$ with $\eta \rightarrow 0$ and $\eta > 0$ can be estimated simultaneously as the estimation of Λ° .

Besides, we analysed the local influence for Λ° by introducing a new sample at the end of the sequence: only a small part of Λ° will be changed. Based on this local property, we proposed an online algorithm (c.f. Algorithm 2 in [3]) for updating the changed part of Λ° , which is more efficient than the offline estimation. The locality of the TV-denoising method motives the adaption of sliding windows in order to deal with

the non-stationary signal by supposing the local stationarity inside the window.

III. ADAPTATION TO SLIDING WINDOWS

In this section, we will present some theoretical elements about the local behaviour of the restoration and adapt the online algorithm to the sliding window.

Let's consider two successive sliding windows $w_i^m = [i, i+m-1]$ and $w_{i+1}^m = [i+1, i+m]$. w_{i+1}^m is indeed w_i^m in which we take off the first point (y_i, τ_i) and add a new point (y_{i+m}, τ_{i+m}) . For the sake of simplicity, $F(u_{\{i, \dots, i+m-1\}}, y_{\{i, \dots, i+m-1\}}, \tau_{\{i, \dots, i+m-1\}}, \lambda)$ is noted $F_i^m(\lambda)$ for the window w_i^m .

We note $u^* = (u_1^*, \dots, u_m^*) = \{v^*, \mathcal{N}^*\} = \arg \min F_i^m$ the restoration of the window w_i^m with a given λ and $\hat{u} = (\hat{u}_1, \dots, \hat{u}_m) = \{\hat{v}, \hat{\mathcal{N}}\} = \arg \min F_{i+1}^m$ for w_{i+1}^m . For the simplify of the presentation, the elements concerning about w_i^m will be noted with the symbol $*$, and that for w_{i+1}^m will be noted with the hat symbol. For example, with a given λ , the segment length set of $u^*(\lambda)$ is $\mathcal{T}^* = \{\mathcal{T}_1^*, \dots, \mathcal{T}_K^*\}$ with $\mathcal{T}_j^* = \sum_{i \in \mathcal{N}_j^*} \tau_i$, and that of $\hat{u}(\lambda)$ is $\hat{\mathcal{T}} = \{\hat{\mathcal{T}}_1, \dots, \hat{\mathcal{T}}_{\hat{K}}\}$ with $\hat{\mathcal{T}}_j = \sum_{i \in \hat{\mathcal{N}}_j} \tau_i$.

A. Influence of slide movement

The update of the restored signal and Λ° need to consider the sliding of index from w_i^m to w_{i+1}^m : a restored signal point is unchanged under the influence of slide movement means $\hat{u}_{i-1} = u_i^*$. In [3], we have shown the following theorem:

Theorem 1. *If there exists an index $j \in \{2, \dots, K^*(\lambda)\}$ such that $\text{sign}(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = \text{sign}(v_{K^*(\lambda)}^*(\lambda) - y_{i+m})$, then the new restoration \hat{u} satisfies $\hat{u}_{i-1}(\lambda) = u_i^*(\lambda)$ for all $i < i_1^{j,*}$.*

The diffusion from the “new” sample y_{i+m} changes only the last part of the restoration of w_i^m up to the junction of two segments whose sign of the variation $v_{j-1}^*(\lambda) - v_j^*(\lambda)$ corresponds to that of $v_{K^*}^*(\lambda) - y_{i+m}$. We can establish a similar theorem (c.f. Theorem 2) about the influence of the first sample y_i for the sliding window w_i^m : the removing of y_i changes only the first part of the restoration.

Theorem 2. *If there exists an index $j \in \{2, \dots, K^*(\lambda)\}$ such that $\text{sign}(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = \text{sign}(v_1^*(\lambda) - y_i)$, then the new restoration \hat{u} satisfies $\hat{u}_{i-1}(\lambda) = u_i^*(\lambda)$ for all $i \geq i_1^{j+1,*}$.*

To sum up, for updating $u^*(\lambda)$ to $\hat{u}(\lambda)$, only the first part and the last part of w_i^m are changed. We introduce the following definitions:

Definition 1. *With $l = i_1^{j,*}$ and $p = i_1^{j+1,*} - 1$ where j is the last segment which satisfies $\text{sign}(v_{j-1}^*(\lambda) - v_j^*(\lambda)) = \text{sign}(v_{K^*}^*(\lambda) - y_{i+m})$ and k is the first segment which satisfies $\text{sign}(v_{k-1}^*(\lambda) - v_k^*(\lambda)) = \text{sign}(v_1^*(\lambda) - y_i)$,*

- (u_l^*, \dots, u_m^*) is called non right-isolated sequence.
- (u_2^*, \dots, u_p^*) is called non left-isolated sequence.
- $(u_{p+1}^*, \dots, u_{l-1}^*)$ is called isolated sequence.

B. Independence between segments

For a given λ (noted $\hat{\lambda}$), the cutting set $\mathcal{N}(\hat{\lambda})$ is given by $\{\lambda^\circ > \hat{\lambda}\}$. In [3], we showed the points inside a segment of $u^*(\hat{\lambda})$ are merged for $\lambda \leq \hat{\lambda}$, and the value of λ provoking the merge is independent to the points outside the segment. By introducing the virtual segment (c.f. Definition 2), the estimation of Λ° can be broken into some sub-problems for each virtual segment, shown in proposition 1. Combining with Theorem 1 and 2, the elements of $\{\lambda^\circ \leq \hat{\lambda}\}$ inside the isolated sequence are not influenced by the sliding movement.

Definition 2. Let $\hat{\lambda}$ and $\epsilon_\lambda > 0$, we have $v^*(\hat{\lambda}) = \{v_1^*(\hat{\lambda}), \dots, v_l^*(\hat{\lambda})\}$, $\mathcal{N}^*(\hat{\lambda}) = \{\mathcal{N}_1, \dots, \mathcal{N}_l\}$, $l = K(\hat{\lambda})$ and $s_i = \text{sign}(v_{i+1}^*(\hat{\lambda}) - v_i^*(\hat{\lambda}))$. For each segment $\{y_{\mathcal{N}_j}, \tau_{\mathcal{N}_j}\}$ with $j = 1, \dots, l$, let $c_i = \frac{\hat{\lambda} + \epsilon_\lambda}{2s_i}$, we introduce the virtual segment $\{y_{\mathcal{N}_j}^+, \tau_{\mathcal{N}_j}^+\}$ where:

- $y_{\mathcal{N}_1}^+ = \{y_{\mathcal{N}_1}, v_1^*(\hat{\lambda}) + c_1\}$, $y_{\mathcal{N}_i}^+ = \{v_i^*(\hat{\lambda}) - c_{i-1}, y_{\mathcal{N}_i}, v_i^*(\hat{\lambda}) + c_i\}$ for $i = 2, \dots, l-1$ and $y_{\mathcal{N}_l}^+ = \{v_l^*(\hat{\lambda}) - c_{l-1}, y_{\mathcal{N}_l}\}$.
- $\tau_{\mathcal{N}_1}^+ = \{\tau_{\mathcal{N}_1}, 1\}$, $\tau_{\mathcal{N}_i}^+ = \{1, \tau_{\mathcal{N}_i}, 1\}$ for $i = 2, \dots, l-1$ and $\tau_{\mathcal{N}_l}^+ = \{1, \tau_{\mathcal{N}_l}\}$.

Proposition 1. Let Λ° the estimation with all the samples $\{y_i, t_i\}_{1, \dots, n}$, $\Lambda_i^\circ = \{\lambda_{i,1}^\circ, \dots\}$ the estimation with i^{th} virtual segment $(y_{\mathcal{N}_i}^+, \tau_{\mathcal{N}_i}^+)$, $\Lambda_1^* = \{\lambda_{i,1}^\circ, \dots, \lambda_{i,n_i-1}^\circ\}$ and $\Lambda_i^* = \{\lambda_{i,2}^\circ, \dots, \lambda_{i,n_i}^\circ\}$ for $i = 2, \dots, l$, we have $\cup_{i=1, \dots, l} \Lambda_i^* = \{\lambda | \lambda \leq \hat{\lambda} \cap \lambda \in \Lambda^\circ\}$.

C. Proposition of algorithms

Let $\Delta g^\circ = (\Delta g(\lambda_1^\circ), \Delta g(\lambda_2^\circ), \dots, \Delta g(\lambda_{n-1}^\circ))$, we note $\Lambda^{\circ,i}$ and $\Delta g^{\circ,i}$, two vectors of size $m-1$, the result of w_{i+1}^m . The results after sliding ($\Lambda^{\circ,i+1}$ and $\Delta g^{\circ,i+1}$) based on w_{i+1}^m can be obtained by updating $\Lambda^{\circ,i}$ and $\Delta g^{\circ,i}$. In this section, we will propose an adaptation of the online algorithm proposed in [3] to the sliding windows.

We will only talk about the online estimation of $\Lambda^{\circ,i+1}$ from $\Lambda^{\circ,i}$ in detail. By following, we note an application of Algorithm 1 in [3] to a given sequence of $\{y\}$ and $\{\tau\}$ as $\Lambda^\circ = \text{DP-TV}(\{y\}, \{\tau\})$.

After choosing $\lambda \geq 0$, called the *cutting point*, the restoration for w_i^m is $u^*(\hat{\lambda})$. We note p and j^* respectively the last point and the last segment of the non left-isolated sequence of $u^*(\hat{\lambda})$, l and j_l^* respectively the first point and the first segment of the non right-isolated sequence of $u^*(\hat{\lambda})$. $\Lambda^{\circ,i+1}$ can be splitted into four parts: (1) Some elements of $\{\lambda^{\circ,i+1} > \hat{\lambda}\}$ are changed following Proposition 1; (2) The non left-isolated sequence $\{\lambda_j^{\circ,i} \leq \hat{\lambda}\}$ for $j \leq p$ is influenced by the removal of the first point y_i ; (3) The non right-isolated sequence $\{\lambda_j^{\circ,i} \leq \hat{\lambda}\}$ for $j \geq l$ is influenced by the new point y_{i+m} ; (4) All $\lambda_j^{\circ,i} \leq \hat{\lambda}$ remains the same for $p < j < l$.

We treat at first the unchanged part of $\Lambda^{\circ,i+1}$: due to the sliding, we have $\lambda_{\{p, \dots, l-2\}}^{\circ,i+1} = \lambda_{\{p+1, \dots, l-1\}}^{\circ,i}$.

For the non right-isolated sequence, let $\epsilon_\lambda > 0$, $\Lambda^a = \lambda_{\{\lambda_{l-1}^{\circ,i+1}, \dots, \lambda_m^{\circ,i+1}\}}^{\circ,i+1}$ can be estimated by $\text{DP-TV}(y_a^+, \tau_a^+)$ with the virtual segment:

- $y_a^+ = \{v_{j_l^*}^*(\hat{\lambda}) - \frac{\hat{\lambda} + \epsilon_\lambda}{2\text{sign}(v_{j_l^*}^* - v_{j_l^*-1}^*)}, y_{\{l+i-1, \dots, m+i\}}\}$
- $\tau_a^+ = \{1, \tau_{\{l+i-1, \dots, m+i\}}\}$

For the non left-isolated sequence, $\Lambda^b = \lambda_{\{\lambda_1^{\circ,i+1}, \dots, \lambda_{p-1}^{\circ,i+1}\}}^{\circ,i+1}$ can be estimated by $\text{DP-TV}(y_b^+, \tau_b^+)$ with the virtual segment:

- $y_b^+ = \{y_{\{i+1, \dots, i+p-1\}}, v_{j_p^*}^*(\hat{\lambda}) + \frac{\hat{\lambda} + \epsilon_\lambda}{2\text{sign}(v_{j_p^*}^* + 1 - v_{j_p^*}^*)}\}$
- $\tau_b^+ = \{\tau_{\{i+1, \dots, i+p-1\}}, 1\}$

The isolated and non-isolated sequences can be assembled in $\Lambda^c = \{\lambda_1^c, \dots, \lambda_m^c\}$ with $\lambda_k^c = \lambda_{k-l+3}^a$ for $k \geq l-1$, $\lambda_k^c = \lambda_k^b$ for $k \leq p-1$ and $\lambda_k^c = \lambda_{k-1}^{\circ,n}$ for $p < k-1 < l$.

It remains $\{\lambda^{\circ,i+1} > \hat{\lambda}\}$. Let $d = \{\lambda^{\circ,i+1} > \hat{\lambda}\} = \{\lambda_k^c > \hat{\lambda}\}$ containing indeed all the last points of $\hat{u}(\hat{\lambda})$'s segments, we can get $\Lambda^d = \lambda_d^{\circ,i+1} = \text{DP-TV}(\hat{u}(\hat{\lambda}), \hat{\tau}(\hat{\lambda}))$.

Finally, $\Lambda^{\circ,i+1}$ can be assembled in the following way:

$$\lambda_k^{\circ,i+1} = \begin{cases} \lambda_k^c, & \text{if } \lambda_k^d \leq \hat{\lambda}. \\ \lambda_{\alpha(i)}^d, & \text{if } \lambda_k^d > \hat{\lambda}. \end{cases} \quad (4)$$

with $\alpha : \mathbb{Z} \rightarrow \mathbb{Z}$ giving the index of i in the vector d .

Algorithm 1 Adaptation of online implementation to sliding window

Require: $(y_i, \dots, y_{i+m-1}), (\tau_i, \dots, \tau_{i+m-1}), (y_{i+m}, \tau_{i+m})$

Require: $\Lambda^{\circ,i}, \Delta g^{\circ,i}, \hat{\lambda}$

Find non right-isolated sequence (l, \dots, m) of $u^*(\hat{\lambda})$

Find non left-isolated sequence $(2, \dots, p)$ of $u^*(\hat{\lambda})$

if $p < l-2$ **then**

$(\Lambda^a, \Delta g^a) = \text{DP-TV}(y_a^+, \tau_a^+)$

$(\Lambda^b, \Delta g^b) = \text{DP-TV}(y_b^+, \tau_b^+)$

$\Lambda^{\circ,i+1} = \{\Lambda_{\{1, \dots, p-1\}}^b, \Lambda_{\{p+1, \dots, l-1\}}^{\circ,i}, \Lambda_{\{2, \dots, m-l+2\}}^a\}$

$\Delta g^{\circ,i+1} = \{\Delta g_{\{1, \dots, p-1\}}^b, \Delta g_{\{p+1, \dots, l-1\}}^{\circ,i},$

$\Delta g_{\{2, \dots, m-l+2\}}^a\}$

$d = \{\lambda^{\circ,n+1} > \hat{\lambda}\}$

$(\lambda_d^{\circ,i+1}, \Delta g_d^{\circ,i+1}) = \text{DP-TV}(\hat{u}(\hat{\lambda}), \hat{\tau}(\hat{\lambda}))$

else

▷ Offline approach

$(\lambda_d^{\circ,i+1}, \tau_d^{\circ,i+1}) = \text{DP-TV}(\{y_i, \tau_i\}_{i+1, \dots, i+m})$

end if

return $\Lambda^{\circ,i+1}$ and $\Delta g^{\circ,i+1}$

The algorithm adapted for the sliding window is gathered in Algorithm 1. With a nice choice of $\hat{\lambda}$, only a small part ($\Lambda^a, \Lambda^b, \Lambda^d$) of the window w_{i+1}^m needs to be updated from w_i^m . The overall complexity for estimating the restoration in a window of size m is in $O(m)$.

IV. PERFORMANCE ANALYSIS

A. Application: noise variance monitoring

We propose a method to detect the shift of noise's variance based on our restoration method. For an observed signal y of size n , we take a sliding window of size m with $m < n$. For each window $w_i^m = [i, i+m-1]$, we apply our denoising algorithm with the automatic choice of λ on the sequence (y_i, \dots, y_{i+m-1}) and $(\tau_i, \dots, \tau_{i+m-1})$ for estimating the restored signal $u^*(\lambda_{\text{ours}}) = (u_1^*, \dots, u_m^*)$. The windowed restoration residual for $j = 1, \dots, m$ is given by

$r_j = y_{j+i-1} - u_j^*(\lambda_{ours})$. The variance of the residual inside w_i^m can be estimated by:

$$(\sigma_i^{m*})^2 = \frac{1}{m-1} \sum_{j=1}^m (r_j - \bar{r}_i)^2 \quad (5)$$

with the mean value of residual $\bar{r}_i = \frac{1}{m} \sum_{j=1}^m r_j$.

For the simulation, the realisation of noise is $\hat{\epsilon}_j = y_j - u_{net,j}$ with u_{net} the original signal and $j = 1, \dots, n$. The estimation of noise variance inside w_i^m is given by $(\hat{\sigma}_i^m)^2 = \frac{1}{m-1} \sum_{j=i}^{i+m-1} (\hat{\epsilon}_j - \bar{\hat{\epsilon}}_i)^2$ with $\bar{\hat{\epsilon}}_i = \frac{1}{m} \sum_{j=i}^{i+m-1} \hat{\epsilon}_j$.

Since $u^*(\lambda_{ours})$ is a good restoration of y , $u^*(\lambda_{ours})$ is similar to u_{net} , which means that σ_i^{m*} is close to $\hat{\sigma}_i^m$ inside each window w_i^m . The noise variance σ_i^{m*} is not available for the real data, so we propose to detect the variance shift in using the residual standard deviation vector $\sigma^{m*} = (\sigma_1^{m*}, \dots, \sigma_{n-m+1}^{m*})$ with σ_i^{m*} obtained by (5) inside w_i^m .

For estimating the noise variance in a window of size m , the complexity is in $O(m)$ with the online implementation of sliding windows, and the overall complexity for monitoring the variance of a signal of size n is in $O((n-m)m)$.

B. Results

For the variance monitoring, the method needs to estimate accurately the noise variance in the ideal case or capture the variation of noise variance in a more realistic case. We use the following criteria to evaluate the performance of variance monitoring:

- Average bias: $\overline{\text{bias}} = \frac{1}{n-m+1} \sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m - \sigma_i^{m*})$.
- Ratio of $\hat{\sigma}^m$ variation explained:

$$\text{RVE} = 1 - \frac{\sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m - \overline{\text{bias}} - \sigma_i^{m*})^2}{\sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m - \bar{\hat{\sigma}}^m)^2} \quad (6)$$

where $\bar{\hat{\sigma}}^m = \frac{1}{n-m+1} \sum_{i=1}^{n-m+1} (\hat{\sigma}_i^m)$. It is indeed the R^2 score between σ^{m*} and $\hat{\sigma}^m$ after adjusting those two items to the same mean value. The range of RVE is between 0 and 1, and $\text{RVE} = 1$ indicates our estimation explains perfectly the variation of σ^{m*} .

We compare our method with Median Absolute Deviation (MAD) following the proposition of [12] and [13] $\sigma = 1.4826 \text{ median}(By - \text{median}(By))$ with $By = \sqrt{2}(y_2 - y_1, \dots, y_n - y_{n-1})$.

At first, we will fix the window length $m = 400$. Our proposition requires a parameter q which stands for the length of the approximation of derivative, see p13 in [3] for more details. We apply our method to a simulated piecewise constant signal with the parameter $\log_{10}(q) = 1$. An example of $y = u_{net} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$ and the noise variance estimations for 400-point windows w_i^{400} and $i = 1, \dots, 1601$ are shown in Figure 1. Both MAD and our method propose an estimation similar to the variance of the noise realisation for every sliding windows. In this example, our method underestimates the variance, but the variation of our estimation inside each sliding window follows well that of $\hat{\sigma}^{400}$. The estimation of MAD propose a precise estimation,

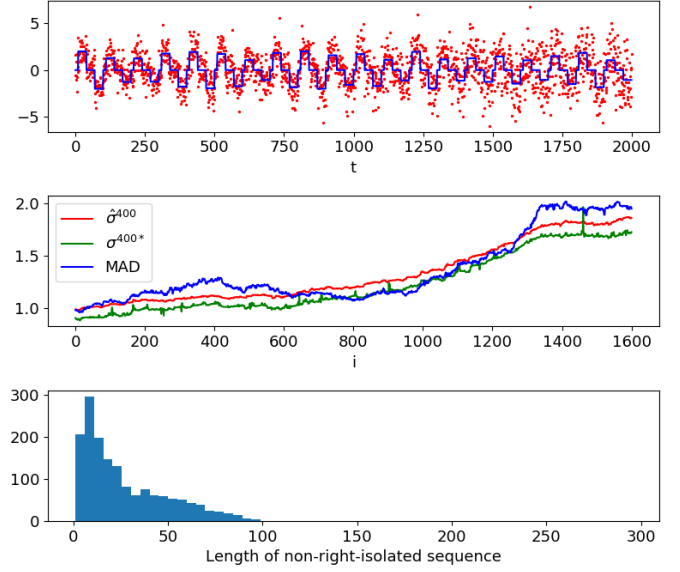


Fig. 1: Up: example of simulated signal: $y = u_{net} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$. Colors : y (red) and u_{net} (blue). Middle: the standard deviation of the noise realisation $\epsilon = y - u_{net}$ (red line), that estimated by our method (green line) and by MAD (blue line) inside every w_i^{400} . Down : histogram of non-right-isolated sequence's length inside each sliding window

but does not follow the variation of $\hat{\sigma}^{400}$. The lengths of non-right-isolated sequence of each window (shown in Figure 1) are all smaller than the window size $m = 400$, which means the influence of new sample remains inside each window and validates our assumption about the local influence.

We tested the performance of MAD and our approach under different noise hypothesis:

- 1) $\mathcal{N}(0, \sigma_a^2(t))$ with $\sigma_a(t) = 1 + 0.0005t$.
- 2) $\mathcal{N}(0, \sigma_b^2(t))$ with $\sigma_b(t) = 1$ for $t \leq 1000$ and $\sigma_b(t) = 1 + 0.001t$ for $t > 1000$.
- 3) $\mathcal{U}(-d(t), d(t))$ with $d(t) = 1 + 0.0005t$.
- 4) $\mathcal{N}(0, \sigma_a^2(t)) + \mathcal{U}(-1, 1)$.

We have done 100 simulations for each type of noise, and the results are gathered in Figure 2. For the most of simulations, our method has $\text{RVE} > 0.95$ and over-performs MAD. The high RVE score indicates the variation of our proposition fits well that of the real values, which allows us to monitor the variation of noise variance for the real signal collected from plants. However, the result of Bias shows our method always underestimates the noise variance, and the bias depends on the type of noise, which means we can not estimate an universal offset for removing this bias in a general case.

The windows length m plays an important role in the restoration and the variance monitoring, and needs to be chosen carefully. A short window can capture the local variation of the noise variance, but the new point (i.e. $(i+m)^{th}$ point) may still have a strong influence for the pass (i.e. $i-1, i-2, \dots$), which limits the restoration performance for the pass points.

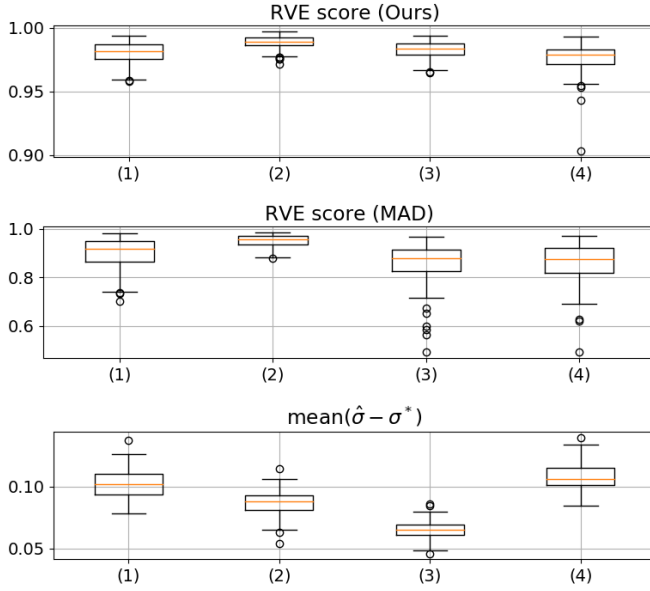


Fig. 2: Result of 100 simulations with different types of noise (1): $\mathcal{N}(0, \sigma_a^2(t))$, (2): $\mathcal{N}(0, \sigma_b^2(t))$, (3): $\mathcal{U}(-d(t), d(t))$ and (4): $\mathcal{N}(0, \sigma_a^2(t)) + \mathcal{U}(-1, 1)$. Up: RVE Score of our method; Middle: RVE score of MAD; Down: Bias of our method.

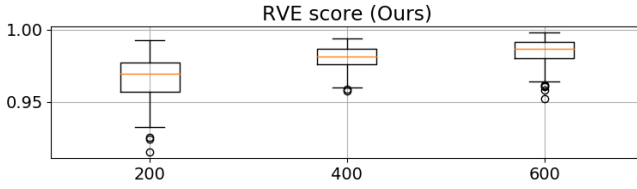


Fig. 3: RVE score of 100 simulations with different window length (m).

We test $m = 200, 400, 600$ with the noise $\epsilon \sim \mathcal{N}(0, \sigma_a^2(t))$, and the results are shown in Figure 3. A longer window provides a better performance of monitoring. But the local variation of σ^2 will be neglected for a long window, and the stationarity hypothesis inside the window is not validated anymore. The choice of m is out of the scope of this article, and it remains an open question.

To sum up, our method can not provide a precise estimation of the noise's variance values, but our estimation follows well that of the real value with an estimation error depending on the type of noise and the shape of original signal.

One of the applications of our method is to detect the shift of noise variance. The residual variance may stay stable (i.e. $\sigma_i^{m*} = \sigma$) during the correct functional period, and the noise variance shift can be detected by a prefixed threshold over σ_i^{m*} comparing to the stable regime: e.g. $\sigma_i^{m*} > 1.2\sigma$ for some successive sliding windows w_i^m .

V. CONCLUSION

In this article, we adapt the TV denoising method proposed in [3] to sliding windows based on the local property of the TV-denoising signal. We believe our adaptive choice of λ works also for the non-stationary signal. We applied the adaption to some signals with time varying noise variance. The simulated results shows that the variance of the restoration residuals follows well that of the noise, and this method can be used to monitor the ground noise variance in real time with a limited computation resource. However, the performance of these methods (both restoration and variance monitoring) depends on the choice of window length m which is our on-going research interest. The application to the real data is also one of on-going works.

ACKNOWLEDGMENT

This work has been supported by the EIPHI Graduate School (ANR-17-EURE-0002).

REFERENCES

- [1] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, 1992.
- [2] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, 1997.
- [3] Z. Liu, M. Perrodin, T. Chambrion, and R. Stoica, "Revisit 1d total variation restoration problem with new real-time algorithms for signal and hyper-parameter estimations," *arXiv preprint arXiv:2012.09481*, 2020.
- [4] R. J. Tibshirani and J. Taylor, "The solution path of the generalized lasso," *The Annals of Statistics*, 2011.
- [5] I. Pollak, A. S. Willsky, and Y. Huang, "Nonlinear evolution equations as fast and exact solvers of estimation problems," *IEEE Trans. on Signal Processing*, 2005.
- [6] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *The annals of Statistics*, 1981.
- [7] D.-H. Shin, R.-H. Park, S. Yang, and J.-H. Jung, "Block-based noise estimation using adaptive gaussian filtering," *IEEE Transactions on Consumer Electronics*, 2005.
- [8] S. Beheshti and M. A. Dahleh, "A new information-theoretic approach to signal denoising and best basis selection," *IEEE Transactions on Signal Processing*, 2005.
- [9] M. Hashemi and S. Beheshti, "Adaptive noise variance estimation in bayesshrink," *IEEE Signal Processing Letters*, 2010.
- [10] J. Han and L. Xu, "A new method for variance estimation of white noise corrupting a signal," in *I2MTC*, 2006.
- [11] S. Pyatykh, J. Hesser, and L. Zheng, "Image noise level estimation by principal component analysis," *IEEE transactions on image processing*, 2012.
- [12] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *biometrika*, 1994.
- [13] S. Sardy and H. Monajemi, "Efficient threshold selection for multivariate total variation denoising," *Journal of Computational and Graphical Statistics*, 2019.